TECHNICAL REPORT ARSCD-TR-80001

APPLICATION OF THE HAAR TRANSFORM TO IR IMAGERY

GARY SIVAK

MARCH 1980

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
FIRE CONTROL AND SMALL CALIBER
WEAPON SYSTEMS LABORATORY
DOVER, NEW JERSEY

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Date Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Technical Report ARSCD-TR-80001	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE *(and Subtitle)*: Application of the Haar Transform to IR Imagery	5. TYPE OF REPORT & PERIOD COVERED: Final
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s): Gary Sivak	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: ARRADCOM, FC&SCWSL Fire Control Division (DRDAR-SCF-I) Dover, NJ 07801	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Proj. No. 12161101A01112
11. CONTROLLING OFFICE NAME AND ADDRESS: ARRADCOM, TSD STINFO (DRDAR-TSS) Dover, NJ 07801	12. REPORT DATE: March 1980 13. NUMBER OF PAGES: 43
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*	15. SECURITY CLASS. *(of this report)*: UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Haar transform
Haar functions
Fast Haar transform digital data analysis
Transformations (mathematics)
Digital image processing

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The author's fast Haar transform algorithm, referenced in a previous report, is analytically shown to elicit signal contrast improvement of digital imagery data. The development of a computer program to measure contrast improvement is explained, and tabular and graphical results of contrast increase are presented. A simulation is also developed to explore the algorithm's ability to improve the signal-to-noise ratio of digital imagery, and tabular results of the decrease of the RMS noise levels and increased signal-to-noise ratios are presented. Finally, the algorithm is run using real infrared imagery data, and picture plots before and after Haar processing are shown.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Date Entered)*

TABLE OF CONTENTS

	Page No.
Introduction	1
Edge Contrast Improvement	2
The Contrast Enhancement Formula	2
Development of Program CONPLT	4
Haar Transform Results of Program CONPLT	7
Signal-to-Noise Improvement Capability	7
RMS Noise Level Scaling	7
Data Presentation to Program STN	11
Haar Transform Results of Program STN	14
The Processing of IR Data	18
Development of the Data Processing Routine: Program FRAME	18
Scanning of the Images	19
Selection of the Spatial Frequency Weighting Coefficients	21
Haar Transform Results of Program FRAME	22
Conclusions	28
References	28
Appendixes	
A Subroutine Haar	29
B Program CONPLT	30
C Program STN	34
D Program FRAME	36
Distribution List	39

TABLES

		Page No.
1	Haar transform signal contrast improvement	8
2	Suggested input parameter selections for signal-to-noise ratio calculation	13
3	Haar transform processed signal strengths	15
4	Haar transform noise propagation (RMS levels)	16
5	Haar transform signal-to-noise ratio improvement	17
6	Minimum and maximum gray level ranges	23

FIGURES

1	Tank target characterization	5
2	Haar transform edge contrast increase: graphical display	9
3	Picture processing scan procedure	20
4	Alabama data base image 1 before Haar processing	24
5	Alabama data base image 1 after Haar processing	25

INTRODUCTION

This report is a follow-on to the investigator's previous report on the development of his fast Haar transform algorithm presented in appendix A (ref 1). After the presentation of a flowchart and Fortran subroutine for the algorithm's implementation, and examples of the algorithm's application to digital signal processing, the following conclusions were drawn which summarize its superiority over the conventional Fast Fourier transform:

1. Between four and five times faster in terms of the computer time required for its implementation.

2. No requirement to generate or store in memory tables of trigonometric values.

3. Mathematical aliasing ambiguities are eliminated.

4. Optimally suited for pulsed data and image analysis and enhancement.

In this report the Fast Haar Transform algorithm is applied to both computer simulated and real imagery data for edge contrast and signal-to-noise improvement. It is theoretically shown to improve edge contrast. Tabular and graphical results of its application to computer simulated data are presented. The algorithm is then shown, by computer simulation, to possess signal-to-noise improvement capability. Three tables present the results in detail of the decrease in the RMS noise level and the signal-to-noise ratio improvement.

In mid-December 1978, the algorithm was applied to process real low air-to-ground IR data to elicit image enhancement. The development of the processing routine, Program FRAME, is explained, and picture plots of imagery before and after Haar processing are included at the end of this report.

EDGE CONTRAST IMPROVEMENT

The Contrast Enhancement Formula

Problem: Calculate the edge contrast improvement capability of the Fast Haar Transform algorithm.

If one works through the equations for N = 4, of the 4-element case of the Haar algorithm in reference 1, the available M-values are: M_1 M_2 and M_3. M_3 govern the highest spatial frequency corresponding to a digital input signal of the form (a, b, a, b). M_2 governs the next highest spatial frequency oscillation corresponding to a digital signal input with values (a, a, b, b). Finally, M_1 govern the strength of the d.c. term.

If one inputs the general signal vector (a, b, c, d) into the Haar algorithm, the 4-element output is

$$\begin{aligned} O_1 &= \tfrac{1}{4}\left[M_1\,(a+b+c+d) + M_2\,(a+b-c-d) + 2\,M_3\,(a-b)\right] \\ O_2 &= \tfrac{1}{4}\left[M_1\,(a+b+c+d) + M_2\,(a+b-c-d) - 2\,M_3\,(a-b)\right] \\ O_3 &= \tfrac{1}{4}\left[M_1\,(a+b+c+d) - M_2\,(a+b-c-d) + 2\,M_3\,(c-d)\right] \\ O_4 &= \tfrac{1}{4}\left[M_1\,(a+b+c+d) - M_2\,(a+b-c-d) - 2\,M_3\,(c-d)\right] \end{aligned} \tag{1}$$

For the particularly high spatial frequency input signal (a, b, a, b), this equation reduces to

$$\begin{aligned} O_1 &= \tfrac{1}{4}\left[2\,M_1\,(b+a) - 2\,M_3\,(b-a)\right] \\ O_2 &= \tfrac{1}{4}\left[2\,M_1\,(b+a) + 2\,M_3\,(b-a)\right] \\ O_3 &= O_1 \\ O_4 &= O_2 \end{aligned} \tag{2}$$

If one evaluates these equations for a flat enhancement characteristic, i.e., no contrast enhancement, by setting the m-values equal to one, the output reduces to the original input data as expected.

If the picture element with the value b is an edge point, and the neighboring one with the value a is in the image background or clutter, the input contrast is defined as

$$C_i = \frac{b-a}{b} \tag{3}$$

or as will be required shortly

$$B = \frac{a}{1 - C_i} \tag{4}$$

Equation 2 is best rewritten

$$O_1 = f - s \tag{5}$$

$$O_2 = f + s$$

where f is the first term of any of equations 2 and s is the second term. The desired output edge contrast is

$$C_o = \frac{O_2 - O_1}{O_2} \tag{6}$$

$$= \frac{2s}{f + s}$$

$$= \frac{2}{1 + \frac{f}{s}}$$

$$= \frac{2}{1 + \frac{M_1}{M_3} \cdot \frac{b + a}{b - a}}$$

Substituting for b from equation 3 results in the theoretical contrast enhancement formula for the particular digital input signal form (a, b, a, b)

$$C_o = \frac{2}{1 + \frac{M_1}{M_3} \cdot \frac{2 - C_i}{C_i}} \tag{7}$$

If these equations are re-evaluated for the lower spatial frequency digital input signal form (a, a, b, b), the resulting output contrast is

$$C_o = \frac{2}{1 + \frac{M_1}{M_2} \cdot \frac{2 - C_i}{C_i}} \tag{8}$$

In general, for either signal, from equations 7 and 8:

$$C_o = \frac{2 \cdot C_i \cdot G}{2 - C_1 + C_1 \cdot G} \tag{9}$$

$$G = \frac{M_i}{M_1} \tag{10}$$

where i = 3 for the high spatial frequency case of equation 7, and 2 for the lower spatial frequency case of equation 8. Note that contrast enhancement is a nonlinear function of the input contrast C_i . However, in the case of extremely low input contrast, the enhancement of a given spatial frequency is linear, and simply equal to the ratio of the m-value of the particular spatial frequency of interest to the m-value of the d.c. term. Thus, contrast has been quantitatively demonstrated to improve under the action of the Haar transform algorithm.

Development of Program CONPLT

Program CONPLT, (see appendix B), is an interactive computer program designed to demonstrate contrast improvement in both tabular and graphic form. As presented in this report, it will quantitatively test the contrast improvement of the author's fast Haar transform algorithm in the 16-element per line scan of the 16 lines of a simulated image of a tank target. The program is not limited solely to testing of the Haar algorithm, but can be applied to any transform or nontransform algorithm of interest by simply tacking it on as a modular subroutine. In fact, the calls in the main program to the various algorithms refer to eight other algorithms, fully documented in program IMAGE of another report, but which, for reasons of space and duplication, were not included with program CONPLT in this report, (ref 2).

The target field is characterized by a 16-by-16 image field comprising a bordering background region of pixels with an assigned level value, and a central signal level region which can be outlined by the graphical drawing of a tank. The characterization of the tank target is shown in figure 1.

The capital letters G, C, T, H, W, and B show the various sections of the tank in the image field and indicate the gun, cupola, turret, hull, wheels (or tread), and the background regions respectively.

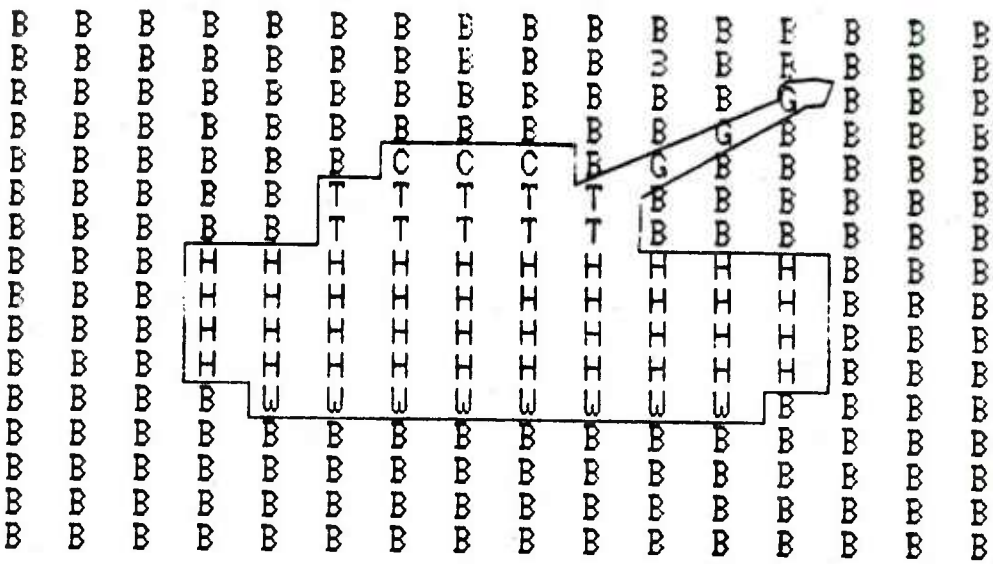

Figure 1. Tank target characterization.

Since this program has been adapted from the coding of program IMAGE, no matrix larger than the 16-by-16 case has been used in this simulation for the same two reasons as for Program IMAGE. First, the fast Haar algorithm requires the dimension N of the input data to be an integer power of 2. Thus, the next larger utilizable image matrix, the 32-by-32 case containing 1024 entry elements is too unwieldy to easily manipulate on paper, both for theoretical studies and numerical value checking. The 16-by-16 image matrix is the largest one that can be fully displayed on the terminal, to output the results of program IMAGE due to the number of alphanumeric characters and spaces available per output line.

The program first requires the insertion of some data. First in the array IC is required the characteristic of the given algorithms, i.e., the holerith letters "H", "V", or "B", to indicate to the user in subsequent interactive questions whether his algorithm of interest scans the image data horizontally, vertically, or in both directions. Second, the array NV contains the number of m-values for a given transform if required.

Then the program asks for the following information:

1. "For printout, plot, or both enter 1, 2, or 3."

2. "Type no. of points not greater than 1001, minimum and maximum contrasts in percent."

If the user has elected to include a plot of output versus input contrast of the given algorithm in his particular run, the program will ask for the following:

"Enter left, right upper, and lower margin portions." By specifying decimal fractions, the user can designate the borders of his graph, for example that the left and right 10 percent, the bottom 30 percent, and the top 20 percent of his display will be blank and surround the graph. This will allow, for example, the placement of captions on hard copies of the graphs.

The program next asks for the background and the signal-to-noise ratio. From the specified bounds for the minimum and maximum values of input contrast, the program computes the signal levels as defined by equation 4.

Finally, the program asks for the particular algorithm desired, and, if required by the algorithm, will request the m-values or any parameters required to control processing. The input images are processed in turn, and the respective tabular and graphical output results displayed.

Haar Transform Results of Program CONPLT

Output signal contrast is calculated specificly by subroutine CTRAST which takes an average over the entire tank of the contrasts of the various features such as the gun and tread. For a detailed discussion of how subroutine CTRAST computes horizontal and vertical edge contrasts of each pixel on each tank component of the output image, see the section on subroutine CTRAST in the referenced author's report on the image enhancement demonstration model.

Contrast improvement is strikingly demonstrated by running the simulated tank target image against the author's Fast Haar Transform algorithm with a set of m-values that gave good pictorial enhancement through program IMAGE, specifically .5, 1, 5, 10, and 15.

Table 1 presents a display of output contrast for the Haar transform with the above-mentioned m-values for input contrasts ranging from zero to 100 percent in increments of 5 percent. The results, as specificly calculated by subroutine CTRAST, are also shown graphically in figure 2 for the same minimum and maximum contrast values. The output contrast rises quickly as a function of the input contrast values to 100 percent and then saturates for high input contrast values.

SIGNAL-TO-NOISE IMPROVEMENT CAPABILITY

RMS Noise Level Scaling

A necessary part of image processing algorithm assessment is the ability to add randomly distributed noise with a specified RMS value to the input imagery data. After processing by the desired algorithms, the intensity of any residual noise can be remeasured. The noise points are generated using the software random number generator whose values range from 0.0 to 1.0, the output of this generator is assumed to be comprised of a uniform random distribution of decimal values.

Problem: Find the value of the positive constant C, which when multiplied by the output of a random number generator, whose values range uniformly from 0.0 to 1.0, will produce a noise distribution with an RMS value of x.

Procedure: The N + one output values of the random number generator are assumed to be uniformly distributed between the values of zero and one, inclusive. Multiplying each value by x, and sorting them produces a series of N + one points P_i with N divisions between them such that:

Table 1. Haar transform signal contrast improvement*

Line	Imput contrast (%)	Output contrast (%)
1	0	0.0
2	5	68.79918
3	10	90.37605
4	15	96.77799
5	20	99.23014
6	25	99.47144
7	30	99.64359
8	35	99.77258
9	40	99.87284
10	45	99.95301
11	50	100.0
12	55	100.0
13	60	100.0
14	65	100.0
15	70	100.0
16	75	100.0
17	80	100.0
18	85	100.0
19	90	100.0
20	95	100.0
21	100	100.0

* m-values = 0.5, 1, 5, 10, and 15.

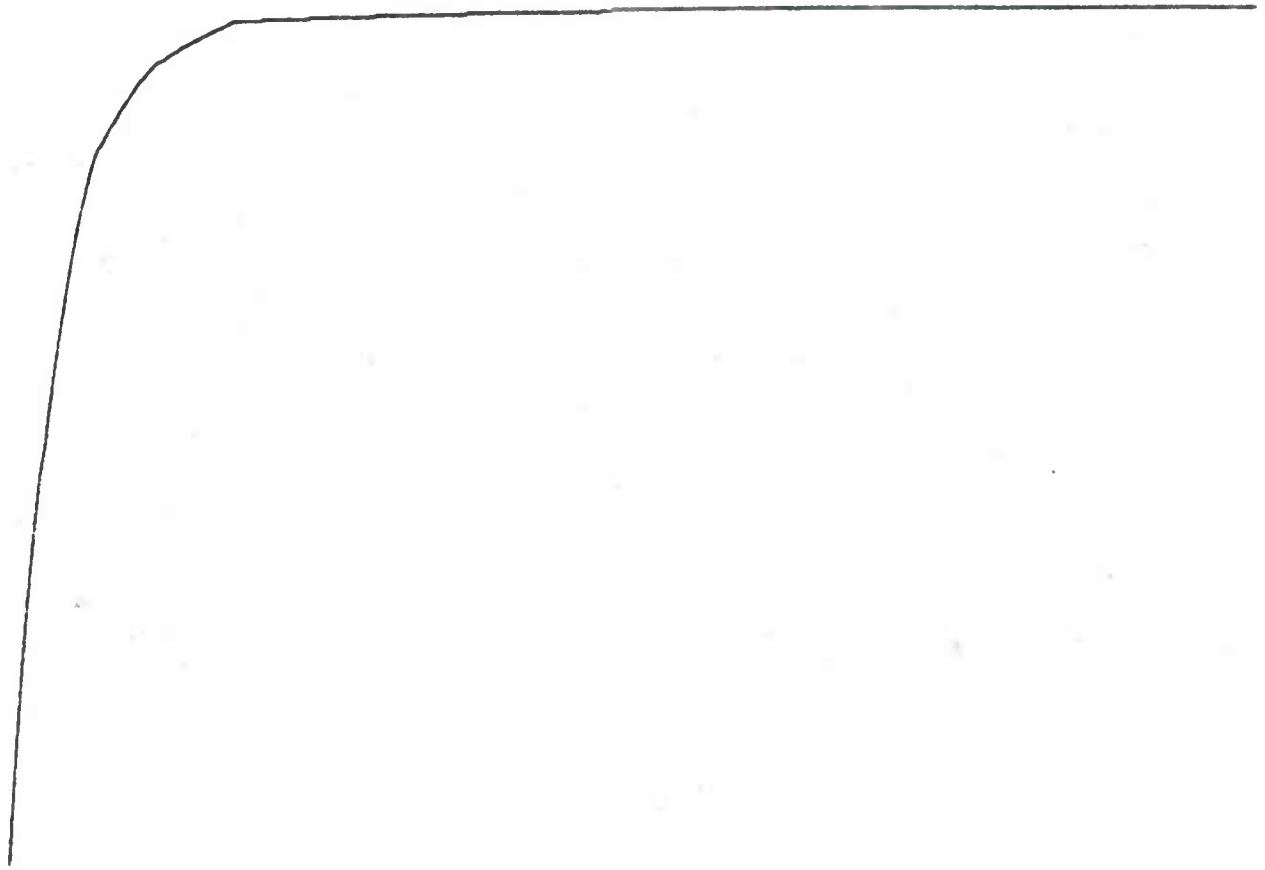

Figure 2. Haar transform edge contrast increase: graphical display.

$$0 \leq P_i \leq x$$
$$0 \leq I \leq N \qquad (11)$$

One must find the RMS value of this collection of noise data points.

$$RMS = \sqrt{\sum_{i=o}^{N} \frac{Pi^2}{N+1}}$$

To calculate the sumation in equation 11 note that if the distribution contains three points and therefore, two divisions between them, i.e., N =2, the points are zero, $\frac{x}{2}$ and x.

The sumation in equation 12 is therefore,

$$S_2 = \left(\frac{1 \cdot x}{2}\right)^2 + \left(\frac{2 \cdot x}{2}\right)^2 \qquad (13)$$

For three divisions between four points the sumation is

$$S_3 = \left(\frac{1 \cdot x}{3}\right)^2 + \left(\frac{2 \cdot x}{3}\right)^2 + \left(\frac{3 \cdot x}{3}\right)^2 \qquad (14)$$

In general, for all N + one points, the sumation for N divisions is:

$$S_N = \left(\frac{1 \cdot x}{N}\right)^2 + \left(\frac{2 \cdot x}{N}\right)^2 + \left(\frac{3 \cdot x}{N}\right)^2 + \cdots + \left(\frac{N \cdot x}{N}\right)^2 \qquad (15)$$

$$= \frac{x^2}{N^2} \cdot \left(1^2 + 2^2 + 3^2 + \cdots + N^2\right)$$

$$= \frac{x^2}{N^2} \cdot \sum_{i=1}^{N} I^2$$

From number theory, a familiar formula yields:

$$\sum_{i}^{N} {}_{=1} I^2 = \frac{N \cdot (N+1) \cdot (2N+1)}{6} \qquad (16)$$

Substituting equation 16 into equation 15 yields:

$$S_N = \frac{X^2}{N} \cdot \frac{(N+1) \cdot (2N+1)}{6} \qquad (17)$$

and therefore from equation 12, the RMS value is:

$$RMS = \sqrt{\frac{X^2}{N} \cdot \frac{2N + 1}{6}} \tag{18}$$

$$= X \cdot \sqrt{\frac{2N + 1}{6N}}$$

This means that multiplying the N + 1 output values of a random number generator by x will produce a random distribution of data with the RMS value of rms as given by equation 18. Dividing by the factor rms gives a data distribution with the RMS value 1. Multiplying by the factor x again, gives a data distribution with the RMS value x, as desired.

Thus, for a random number generator producing N positive values bounded by the value 1, the constant c that when multiplied by this output will produce a noise distribution with the RMS value of x is

$$C = \frac{x^2}{RMS} \tag{19}$$

where RMS is the quantity as computed in equation 18. If the random number generator is not sign-biased, i.e., it ranges between -1 and 1, the RMS noise level scaling constant then is

$$C = \frac{2x^2}{RMS} \tag{20}$$

Data Presentation to Program STN

Program STN is an interactive modular subroutine package which enables the user to test the signal-to-noise improvement capability of up to 16 sets of control parameters or m-values for any desired set of digital image processing algorithms of interest. The simulated imagery data employed for the signal-to-noise measurements is the same 16-by-16 field containing the tank target used in program COMPLT.

First, to measure post-processing signal strengths, the background is set to zero, and a signal value is read into the tank portion of the target field. The target field is then processed by the given algorithm of interest, and the average signal value read out through subroutine APSIG. Then on a second pass, random noise data with a user-specified RMS value is read into the target field which is then processed by the given algorithm. The RMS value of the post-processing 16-by-16 noise pattern is then read out through subroutine PRMS, and can, if desired, be averaged over any number N of runs. The resulting average RMS noise level is then divided into the signal value to produce the new signal-to-noise ratio after processing by the desired algorithm and control parameters.

The program first expects, in the data array NV, the number of m-values or enhancement factors, if required, for each given algorithm the user is interested in testing. The array IC contains holerith codes "H", "V", and "B", specifying for the user's convenience during subsequent queries during data insertion, whether the m-values referred to govern enhancement in the horizontal, vertical, or both directions.

Into the array SIGI are read in seven chosen input signal levels: 9, 10, 11, 12, 14, 16, and 18. Into the 6-element input noise array XMI are read in the input noise levels: 2, 3, 4, 5, 6, and 10. The array SNI contains the chosen input signal-to-noise ratio values. Table 2 shows the array positions and values of the input signal and noise levels that are utilized to obtain the input and also the post-processing signal-to-noise ratios. These array positions are contained in the array IWORK. For example, the first two entries, 1 and 5 mean that the first input or output signal-to-noise ratios are computed from the first input or output signal value, and the fifth input or output noise value respectively. Last, the assigned variables NSIG, NNOI, and NSTN designate to the program's do loops the maximum number of signal, noise, and signal-to-noise values permitted. The program then asks for the minimum and maximum algorithm numbers, so it can test the proper ones, example algorithms four through seven. The currently available algorithms are the nine discussed in the author's referenced report on his image enhancement demonstration model (ref 2).

The user must then indicate the number of data test sets per algorithm, i.e., how many sets of m-values or control parameters, and the number of times the random noise field is to be run per test set. Finally, assuming that they are needed, the program requests, for each data test set of a given algorithm, the required m-values or control parameters against which signal-to-noise improvement will be tested. At this point, execution begins.

Table 2. Signal input parameter selections for signal-to-noise ratio calculation

Signal array		Noise array		Input s/n
position	value	position	value	ratio
1	9	5	6	1.5
2	10	4	5	2.0
2	10	3	4	2.5
1	9	2	3	3.0
5	14	3	4	3.5
6	16	3	4	4.0
7	18	3	4	4.5
2	10	1	2	5.0
3	11	1	2	5.5
4	12	1	2	6.0

Haar Transform Results of Program STN

Tables 3, 4, and 5 summarize the results of signal-to-noise improvement of the tank target data field by the author's Fast Haar Transform algorithm. A one-dimensional horizontal scan with each 16-pixel line per transform was employed. The input signal-to-noise before processing ranged from 1.5 to 6.0 by increments of 0.5. Table 3 charts the signal values after processing, table 4 the noise, and table 5 the increases in signal-to-noise ratio. Four sets of m-values were utilized to obtain the signal-to-noise increases by suppressing the highest spatial frequencies that contain the quickly varying noise spikes. Notice that signal-to-noise can be increased by more than a factor of two by the Fast Haar transform algorithm.

Table 3. Haar Transform processed signal strengths

Input	set 1[a]	set 2[b]	set 3[c]	set 4[d]
9.0	8.01563	7.17188	4.71094	4.67578
10.0	8.90625	7.96875	5.23438	5.19531
11.0	9.79688	8.76563	5.75781	5.71484
12.0	10.68750	9.56250	6.28125	6.23430
14.0	12.46875	11.15625	7.32813	7.27344
16.0	14.25	12.75	8.375	8.3125
18.0	16.03125	14.34375	9.42188	9.35156

a. m-values: 1, 1, 1, 1, 0

b. m-values: 1, 1, 1, 0, 0

c. m-values: 1, 1, 0, 0, 0

d. m-values: 1, 0, 0, 0, 0

Table 4. Haar Transform noise propagation (RMS levels)

Input	Set 1 [a]	Set 2 [b]	Set 3 [c]	Set 4 [d]
2.0	1.37005	1.01052	0.71829	0.43636
3.0	2.02438	1.46790	0.93893	0.64455
4.0	2.75566	1.93931	1.39975	0.97335
5.0	3.37843	2.43969	1.74338	1.33079
6.0	4.22668	2.90235	2.18670	1.40853
10.0	6.83238	5.00663	3.34035	2.04135

---

a. m-values: 1, 1, 1, 1, 0

b. m-values: 1, 1, 1, 0, 0

c. m-values: 1, 1, 0, 0, 0

d. m-values: 1, 0, 0, 0, 0

Table 5. Haar Transform signal-to-noise ratio improvement

Input	Set 1[a]	Set 2[b]	Set 3[c]	Set 4[d]
1.5	1.89643	2.47106	2.15436	3.31961
2.0	2.63621	3.26630	3.00244	3.90394
2.5	3.23198	4.10907	3.73951	5.33754
3.0	3.95954	4.88581	5.01737	7.25438
3.5	4.52477	5.75269	5.23531	7.47255
4.0	5.17117	6.57450	5.98322	8.54006
4.5	5.81756	7.39632	6.73112	9.60757
5.0	6.50067	7.88577	7.28726	11.90608
5.5	7.15074	8.67435	8.01598	13.09669
6.0	7.80081	9.46292	8.74471	14.28729

---

a. m-values: 1, 1, 1, 1, 0

b. m-values: 1, 1, 1, 0, 0

c. m-values: 1, 1, 0, 0, 0

d. m-values: 1, 0, 0, 0, 0

THE PROCESSING OF IR DATA

Development of the Data Processing Routine: Program FRAME

Program FRAME is the name of the job designed to test the author's Fast Haar Transform algorithm against the 1976 Alabama data base Texas Instruments (T.I.) Thermoscope low air-to-ground digital IR imagery data. The digital magnetic tape of the images has 43 image frames, each containing one or several targets, such as the fronts or sides to a tank, armored personnel carrier, or jeep. Each image stores 420 lines of information with 335 pixels per line. The images are described by the vendor as having a relatively clean quality.

Since the Fast Haar Transform algorithm requires N, the number of data elements inputed, to be an integer power of 2, the output pictures that have been generated are not 335 by 420 frames. Only the central region of the Alabama data base images, a 320 by 416 array has been processed, by dropping 4 exterior horizontal lines, 2 above and 2 below, and 15 columns, 7 on the left and 8 on the right edge of the picture. This allows for sectioning of the image into blocks, the dimensions of which are integer powers of 2.

Immediately following the data statements that control the image scan, and the spatial frequency weights discussed in the next two sections, the program requires the input of the number of lines to drop into the picture before starting, and the number of columns to shift over to create the central 320 by 416 region for sectioning and processing. The user then inputs the desired level of any noise to be added to the imagery. This is accomplished by the random number generator, the output of which is scaled according to equation 20. Then, each of the 43 images is scanned in turn, the results of which are written in hexadecimal to a CDC compatible output digital tape. This tape of Haar Transform processed imagery is then reformatted to produce a second output tape compatible with PDP computer equipment formats for subsequent dumping to an electrostatic plotter. This plotter, by printing specified dot densities per unit area as a function of gray level, produces an acceptable display of a multitone picture on a bilevel device, which is available onsite versus a more expensive analog display Cathode Ray Tube (CRT) (ref. 3).

Scanning of the Images

As shown in figure 3, each image has been divided into eight sections for scanning, because the permissible memory allowed for processing on the ARRADCOM CDC 6500 computer cannot contain the entire picture frame array. The entire picture is divided, from top down into three 128 line by 320 point per line horizontal belts, and a remaining 32 line belt at the bottom. Each horizontal belt is then divided into two sections each, the first one extending from the left over 256 points, and the second, 64 points to the right edge of the image. There are therefore eight sections in each image. On the left from the top down there are three 128 line by 256 point per line blocks, and at the bottom a 32 line by 256 point per line block. On the right, there are from top down, three 128 line by 64 point per line blocks and at the bottom, a 32 line by 64 point per line block. The upper left block is block one, and block two is to the right of one. Proceeding down through the image frame, the remaining six blocks are numbered consecutively.

The picture was arbitrarily sliced into these eight regions for two reasons. First, the Haar Transform algorithm requires the number of samples input, and therefore the horizontal and vertical block dimensions, to be an integer power of two. Second, the allotted, maximum memory portion permitted by the CDC 6500 machine is 240,000 octal core spaces. This is 40,960 decimal pixel locations, the size of a 256 by 320 array of values. Bearing in mind that some of this memory portion must also be designated to contain the Fortran programming and compilation, the logical choice amount of data storage is a 128 line by 320 point per line swath.

The processing per picture is done on one of the 4 horizontal belts at a time, from top to bottom, i.e., each belt is copied to the input magnetic tape, replaced by the algorithm with the new post-processing pixel data, and then copied to the output digital magnetic tape.

Within each belt, the extreme left section is processed first, and then the right section. In terms of the program parameters, the particular horizontal belt being executed is governed by the doloop parameter index IBELT, which ranges from one to four. Within each belt, the particular block or section currently being processed is designated by the doloop parameter variable ISEC, which ranges from one to two. The data array NPH contains the number of points in the horizontal direction of each section, and NPV contains the number of lines in each of the four data belts.

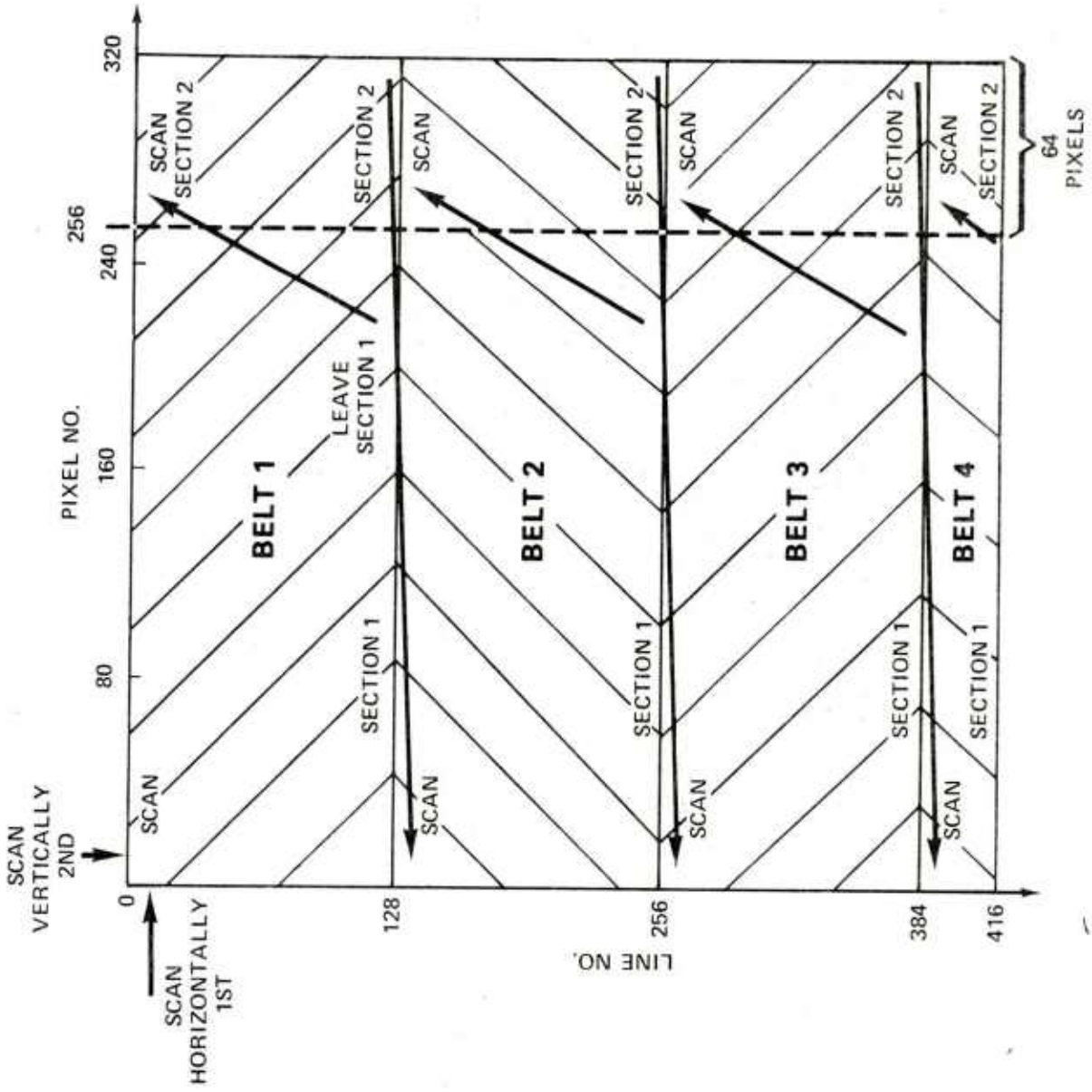

Figure 3. Picture processing scan procedure.

The program allows for further division of each section through the data variables NBH and NBV. They stand for the number of subblocks, in the horizontal and vertical directions into which each of the sections can be subdivided. For the particular imagery discussed in this report, they are simply set to one. In the program, NPBH and NPBV are horizontal and vertical dimensions of each subblock.

Each subblock is scanned by the Fast Haar algorithm, first by horizontal lines down through the NROWB rows, then by vertical columns across the NCOLS number of columns. The Haara Transform contrast improvement for enhancement is accomplished by the Fortran variable arrays EHOR and EVERT which are assigned according to belt and section from the data arrays EH and EV for horizontal and vertical scanning respectively.

Selection of the Spatial Frequency Weighting Coefficients

The horizontal and vertical Haar transform enhancement factors are contained in the input data arrays EH and EV respectively. The EH array contains two sets of horizontal m-values, corresponding to the two sections in each belt. Similarly, the array EV contains four sets of m-values, one for the vertical extent of each data belt. For the two sections of each horizontal belt, the number of points N required by the Haar transform for horizontal scanning are 256 and 64. For the four belts, the number of points required by the Haar transform for vertical scanning are 128, and 32.

As there is no apriori theory to govern the selection of the Haar transform weighting coefficients, a decision was made to produce contrast enhancement by keeping the four highest spatial frequencies, and by suppressing the others down to 25 percent of their original value. With those selections, the original low-contrast data ranging from 2730 to 2890 produced outputs with pixel values ranging from 0 up to about 700.

For the array EH, for N = 256, there are NU + 1 or nine m-values. From the highest spatial frequency down they govern 128 cycles of oscillation of two points each, 64 cycles of oscillation of 4 points each, 32 cycles of oscillation of 8 points each, down to an eighth of one cycle of oscillation over 256 points, and finally the d.c. term. For N = 64 there are seven m-values ranging from 32 cycles of 2 points each down to 1 cycle of 64 points for the sixth, and the d.c. term. Similarly, for the three 128-line belts, there are eight m-values, and six m-values for the 32 line belt.

Thus for N = 256, for the first data set, in the array EH, the m-values are: 0.25, 0.25, 0.25, 0.25, 0.25, 1.0, 1.0, 1.0, and 1.0. In the second set, for N = 64, the m-values are: 0.25, 0.25, 0.25, and 1.0, 1.0, 1.0, and 1.0. Two zeros serve as fillers at the end of the N = 64 portion of the EH array.

For the three 128-line belts, the m-values in the EV array are: four 0.25's and four 1's again with a trailing zero as a space holder. For the last 32 line belt of the EV array, the m-values are two 0.25's and four 1's, again with three trailing zeros at the end as space fillers. In each case the m-values start at the same highest possible spatial frequency, $\frac{N}{2}$ cycles of square wave oscillation of two points each, and spread down by powers of two to one cycle of oscillation over N values, and finally the d.c. term.

Haar Transform Results of Program FRAME

On Monday, December 11, 1978, the 43 Alabama data base IR images were run against the Fast Haar Transform algorithm. As described before, the algorithm scanned each of the eight section of the images, first horizontally, to enhance vertical edges, then vertically, to enhance horizontal lines. In any given data set to the algorithm, the highest four spatial frequency components were preserved, while all others were suppressed by a factor of four. This produced edge contrast improvement with G equal to 4, of equations 9 and 10 as follows:

$$C_o = \frac{8C_i}{2 + 3C_i} \qquad (20)$$

Thus for low contrast areas of the image, contrast was improved by a factor of 4,. Improvement was slightly less for initially higher contrast areas of the imagery data.

With the removal of the enormous d.c. term from the imagery by transform, a term 17 times the original total brightness variation of 160, and with the algorithm's enhancement of gray level values, the output grey levels ranged from about 0 to 700. Table 6 is a chart presenting the minimum and maximum gray-level ranges of each of the 43 images before and after processing.

Figures 4 and 5 are electrostatic plot displays of the first Alabama Data Base Ir image before and after Haar processing. The target is a tank side located 6/8ths of the way down from the top of the image, and 2/7ths of the way across from left to right. It is roughly centered on the plot in a 64-by-64 region bounded on the left and right by pixels 32 and 96, and above and below by lines 232 and 296.

Table 6. Minimum and maximum gray level ranges

No.	Before processing		After processing	
	Gray level		Gray level	
	min	max	min	max
1	2730	2877	9	711
2	2750	2897	5	721
3	2730	2890	13	766
4	2730	2890	24	734
5	2610	2770	16	756
6	2610	2770	6	705
7	2590	2750	8	707
8	2610	2770	16	706
9	2790	2870	24	726
10	2730	2875	16	733
11	2670	2830	4	702
12	2690	2850	20	727
13	2777	2910	20	735
14	2770	2850	30	717
15	2610	2770	18	706
16	2670	2830	22	724
17	2730	2890	17	728
18	2590	2750	-1	700
19	2790	2862	33	715
20	2670	2830	23	716
21	2710	2790	21	723
22	2770	2850	31	714
23	2730	2890	-4	759
24	2730	2834	16	724
25	2759	2890	11	751
26	2670	2830	16	707
27	2610	2770	5	725
28	2610	2770	0	729
29	2590	2750	3	729
30	2730	2890	7	725
31	2730	2890	11	726
32	2742	2870	6	742
33	2775	2910	25	734
34	2756	2910	20	752
35	2764	2890	6	727
36	2780	2850	24	721
37	2670	2830	11	728
38	2650	2810	10	735
39	2650	2810	19	723
40	2730	2810	33	703
41	2730	2810	33	706
42	2710	2790	22	706
43	2751	2810	34	704

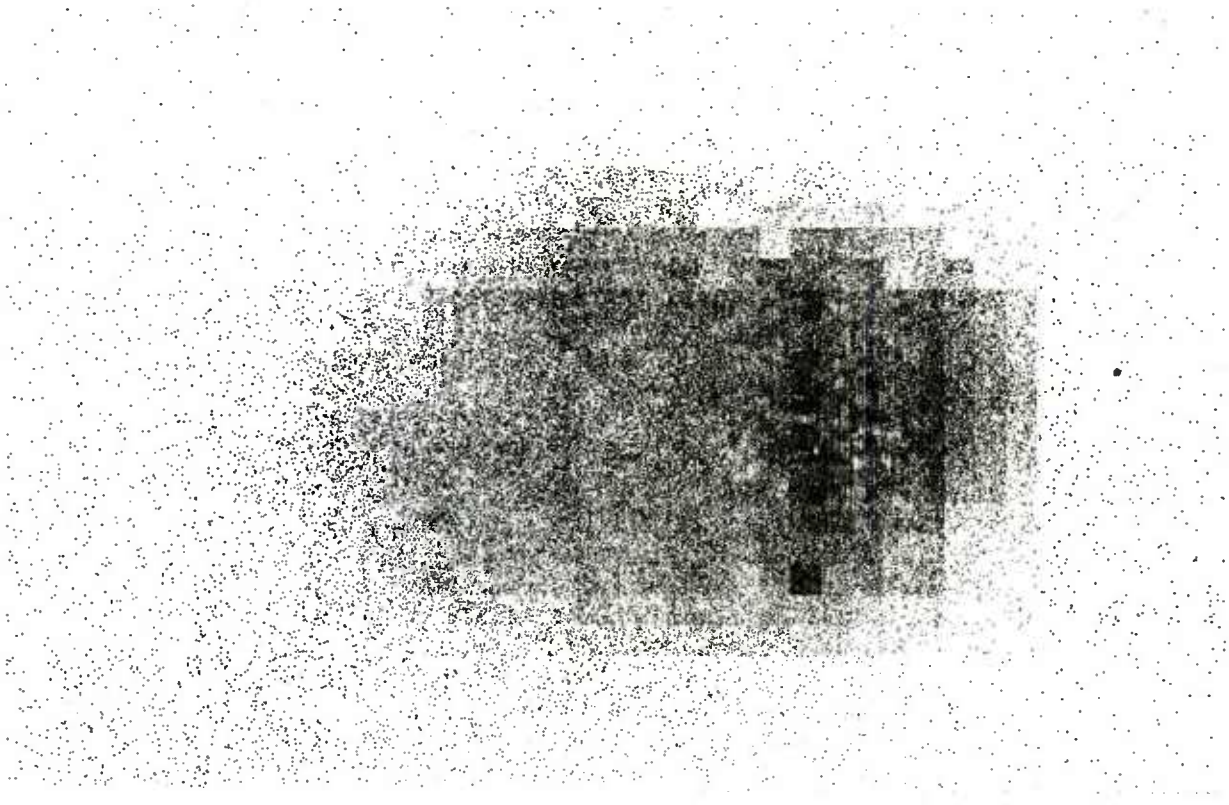

Figure 4. Alabama data base image 1 before Haar processing.

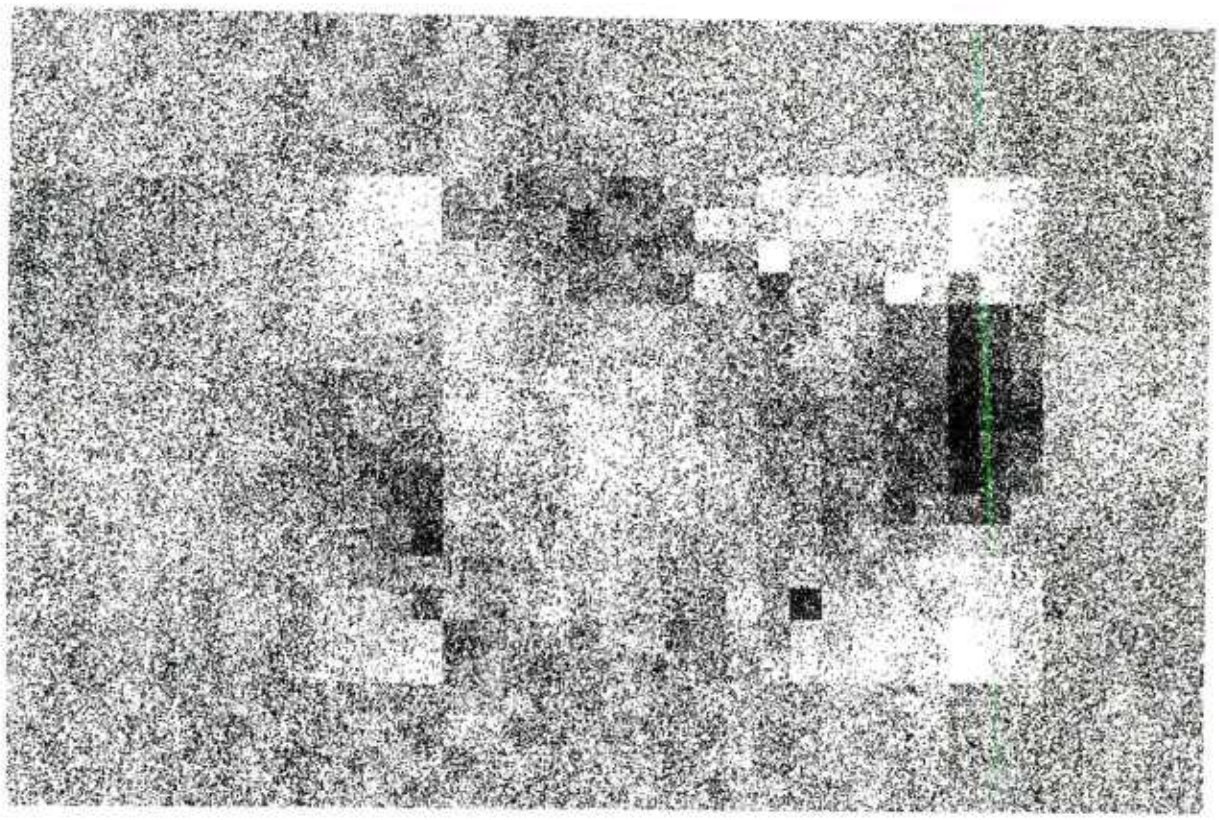

Figure 5. Alabama data base image 1 after Haar processing.

The electrostatic plotter used to display the results can print a maximum of 200 dots per inch on the paper. The gray level for each pixel is represented by randomly filling a small square on the paper corresponding to the pixel location with a given number of dots as a nonlinear function of gray level. As determined by photometer measurements, the number of dots as a function of gray level is not linear and is best given by:

$$\text{DOTS} = \text{DOTMAX} \cdot \left(\frac{\text{Graylevel - Min. Gray level}}{\text{Max. Graylevel - Min. Gray level}} \right)^{2.6} \tag{21}$$

With the large d.c. pedestle removed from the processed images, and a contrast increase of four over the dynamic range, one expects at least a fourfold increase in the display quality of the images. One might think that the large number of pixels in the original 420 by 335 imagery would be indicative of good resolution and therefore high video information content. However, the fact that the original imagery looks as if it is composed of approximate tankshaped blobs is due to the intrinsic low quality of the original imagery, the digitized version which was obtained in a roundabout manner. The high pixel content did not emerge from the use of a high resolution sensor, but simply from the digitization of polaroid photos of the output display of the original data.

Although the Thermoscope imagery obtained by time delay signal integration is relatively clean, it is blurred and has relatively low resulution because of the quality of the IR optics and sensor used. Transitions therefore, are not sharp because the target edges are blurred and not clearly defined. Since the digitized versions of the images were obtained from the photos of the data, the many pixels simply do not have a high information content. Although the Haar algorithm will sharpen the edges of low contrast data and will supress the noise of noisy imagery where little information content is provided to start with, even improvement by an order of magnitude will not elicit much observable difference.

The outputs of the 11 December 1978 application of the Haar algorithm to the Alabama data base imagery data resulted in outputs comprised of three vertical bands, a light one on the left, a gray one in the middle, and a black one on the right. It was then discovered that the computer program designed by MISD to translate the tape of 60-bit CDC binary hexadecimal words into an output tape of 8-bit PDP words was not successful. Dumps of the data from the two tapes did not produce identical pixel values. It was therefore decided to circumvent the problem of computer code translation by rerunning the Haar algorithm against the Alabama data base imagery directly on the PDP machine, thus eliminating the need to translate to and back from CDC binary hex codes.

Only the 64-by-64 region of image No. 1, in which the target of interest is centered, was run against the Haar algorithm because of the constraints on the alloted memory allowed for a given task by the RSX11M operating system.

For the 64-by-64 array, N = 64, and there are therefore seven weighting factors available to produce the Haar output of figure 5. Starting with the d.c. term, they were chosen to be: 0.1, 0.0, 0.0, 1.0, 1.0, 1.0, and 1.0 in order to produce both suppression of d.c. and low frequency and edge enhancement.

The quality of the resulting pictures was less than desirable for four reasons. First, as previously mentioned, the original imagery was of poor quality. Second, the algorithm unfortunately introduces distracting gray into the picture background by trying to enhance background information in addition to the target. Third, a 64-by-64 array does not provide much resolution. And fourth, the diminution and suppression of discontinuous low spatial frequency components from a smooth image renders the data in the discontinuous form of equation 2.3.

High spatial frequencies = smooth - low spatial frequencies (22)

The prominence of discontinuous square wave components in the images, and the fact of low resolution impart a distracting checkerboard pattern to the enhanced imagery. However, as the theory suggests, one does notice that the algorithm did attempt target edge enhancement, because the target edges are indeed darker than the central region.

CONCLUSIONS

The Haar transform algorithm is shown to be theoretically useful for improving edge feature contrast of images, and for improving target signal-to-noise ratio. It fails, however, to suitably enhance IR imagery data for the display of tactical targets of interest.

REFERENCES

1. Gary Sivak, "The Haar Transform: Its Theory and Computer Implementation," April 79, ARRADCOM Report No. ARSCD-TR-78005.

2. Gary Sivak, "Video Signal Processing For Image Enhancement: A Demonstration Model," (being published).

3. J. F. Jarvis, C. N. Judice, W. H. Ninke, "A Survey of Techniques for the Display of Continuous Tone Pictures on Bilevel Displays," Bell Labs Journal of Computer Graphics and Image Processing, Vol. 5, No. 1, March 1976, pp 13-14.

APPENDIX A. SUBROUTINE HAAR

```
      SUBROUTINE HAAR(Q,E,NU)
      DIMENSION T(16),Q(30),E(5)
      N =2**NU
      NO =N
      ME = NU+2
      IP = (-2)*N
      IR = 0
    5 NO = NO/2
      IP = IP + 4*NO
      IR = IR + 2*NO
      ME = ME-1
      DO 6 J =1,NO
      L = 2*J
      K = L-1
      NOJ = NO+J
      T(NOJ) = E(ME)*(Q(IP+K)-Q(IP+L))
    6 Q(IR+J) =Q(IP+K)+Q(IP+L)
      IF (NO.GT.2) GO TO 5
      T(2) =E(2)*(Q(IR+1)-Q(IR+2))
      T(1) =E(1)*(Q(IR+1)+Q(IR+2))
      Q(1) =(T(1)+T(2))/FLOAT(N)
      Q(2) =(T(1)-T(2))/FLOAT(N)
      IP= -1
      IR = 0
    7 N2 = NO
      X = FLOAT(N2)/FLOAT(N)
      NO = NO*2
      IP = IP + NO/4
      IR = IR + NO/2
      DO 8 J =1,NO
      ND = (J+1)/2
      ID = N2 + ND
      Y = (-1.0)**(J+1)*X
    8 Q(IR+J) = Q(IP+ND)+Y*T(ID)
      IF (NO.LT.N) GO TO 7
      RETURN
      END
```

APPENDIX B. PROGRAM CONPLT

```
      PROGRAM CONPLT(INPUT,OUTPUT,TAPE61=100,TAPE62=100)
      DIMENSION X(1001),Y(1001),HM(8),VM(8),IC(9),NV(9)
      DATA IC/"H","V","B","H",'H',"H","B","H","B"/
      DATA NV/5,5,5,3,4,8,8,8,8/
      CALL INITT(30)
      CALL TERM(2,4096)
      PRINT *,"FOR PRINTOUT, PLOT, OR BOTH, ENTER 1, 2, OR 3. "
      READ *,IOP
      PRINT *,"TYPE NO. POINTS NOT.GT.1001, MIN & MAX CONTRASTS IN % "
      READ *,NP, XMIN, XMAX
      IF(IOP .EQ. 1) GO TO 1
      PRINT *,"ENTER LEFT, RIGHT, UPPER AND LOWER MARGIN PORTIONS. "
      READ *,XLM,RM,TM,BM
    1 PRINT *,"ENTER BACKGROUND AND SIGNAL-TO-NOISE RATIO. "
      READ *,B,STN
      PRINT *,"ENTER ALGORITHM NO. "
    2 READ *,IALG
      IF(IALG .LE. 9) GO TO 3
      PRINT *,"WRONG ALGORITHM NO., ",IALG,", TRY AGAIN. "
      GO TO 2
    3 N = NV(IALG)
      IF(IC(IALG) .EQ. "V") GO TO 5
      I1 = 1
      IF(IC(IALG).EQ."H") I2 = 1
      IF(IC(IALG).EQ."B") I2 = 2
      GO TO 6
    5 I1 = 2
      I2 = 2
    6 DO 7 IKIND = I1,I2
      PRINT *,"ENTER ",N," ALG. ",IALG,IC(IALG),"M VALUES. "
      IF(IKIND.EQ.1) READ *,(HM(J),J=1,N)
    7 IF(IKIND.EQ.2) READ *,(VM(J),J=1,N)
   13 DX = (XMAX - XMIN ) / FLOAT(NP-1)
      YMIN = 1.E100
      YMAX = -1.*YMIN
      I = 0
   14 I = I + 1
      X(I) = XMIN + DX * FLOAT(I-1)
      XI = X(I)
      CALL DIRK(XI,YI,B,STN,IALG,IC,HM,VM)
      Y(I) = YI
      IF ( Y(I) .LT. YMIN ) YMIN = Y(I)
      IF ( Y(I) .GT. YMAX ) YMAX = Y(I)
      IF(I.LT.NP) GO TO 14
      PRINT *,"DONE, YMIN = ",YMIN,", YMAX= ",YMAX
      YMAX = YMAX + 1.E-14
      YMIN = YMIN - 1.E-14
      HH = (XMAX - XMIN ) / (1. - XLM - RM)
```

```
      XMIN = XMIN - HH * XLM
      XMAX = XMAX + HH * RM
      HV = (YMAX - YMIN ) / (1. - TM - BM)
      YMAX = YMAX + HV * TM
      YMIN = YMIN - HV * BM
      READ *,DUMMY
      IF ( IOP .EQ. 2) GO TO 15
      PRINT 280
  280 FORMAT(" LINE          X                    Y")
      DO 16 J = 1,I
      PRINT 310,J,X(J),Y(J)
  310 FORMAT(1H ,I3,2(6X,F10.5))
      IF (MOD(J,30).NE. 0) GO TO 16
      READ *,DUMMY
      PRINT 280
   16 CONTINUE
      READ *,DUMMY
      IF ( IOP .EQ. 1) STOP
   15 CALL DWINDO(XMIN,XMAX,YMIN,YMAX)
      CALL MOVEA ( X(1), Y(1) )
      DO18 J = 2,I
   18 CALL DRAWA ( X(J), Y(J) )
      CALL ANMODE
      READ *,DUMMY
      END
      SUBROUTINE DIRK(XI,YI,B,STN,IALG,IC,HM,VM)
      DIMENSION IC(9),HM(8),VM(8),XIN(16,16),G(12)
      S = B/ (1.-XI/100.+1.E-14)
      RMS = S/ STN
      DO 6 I=1,16
      DO 6J=1,16
    6 XIN(I,J)=B
      DO 7 I=1,3
      XIN(2+I,14-I)=S
    7 XIN(5,6+I)=S
      DO 8 I=6,7
      DO 8 J=6,10
    8 XIN(I,J)=S
      DO 9 I=8,11
      DO 9 J=4,13
    9 XIN(I,J)=S
      DO 10 J=5,12
   10 XIN(12,J)=S
      SQ = RMS*SQRT((2.*16.+1.)/(6*16.))
      C=RMS*RMS/SQ*2.0
      DO 11 I=1,16
      DO 11 J=1,16
   11 XIN(I,J)=XIN(I,J)+(RANF(9)-0.5)*C
      IF (IALG.EQ.1) CALL HAR16H(XIN,HM)
      IF (IALG.EQ.2) CALL HAR16V(XIN,VM)
      IF (IALG.EQ.3) CALL HAR16HV(XIN,HM,VM)
```

```
      IF (IALG.EQ.4) CALL HAR4H(XIN,HM)
      IF (IALG.EQ.5) CALL HAD4H(XIN,HM)
      IF (IALG.EQ.6) CALL FT16H(XIN,HM)
      IF (IALG.EQ.7) CALL FT16D2(XIN,HM,VM)
      IF (IALG.EQ.8) CALL FFTH(XIN,HM)
      IF (IALG.EQ.9) CALL FFTHV(XIN,HM,VM)
      CALL CTRAST(XIN,G)
      IF(IC(IALG) .EQ. "H") YI = G(11)
      IF(IC(IALG).EQ."V") YI = G(12)
      IF(IC(IALG).EQ."B") YI = (G(11) + G(12))/2.0
      RETURN
      END
C DIRK CY2, D100,1350, TACKON, EDIT, D1220*EOR.
      SUBROUTINE CTRAST(R,G)
      DIMENSION R(16,16),G(12)
      DO 2 I = 1,16
      DO 2 J=1,16
    2 IF(R(I,J).LE.0.) R(I,J)=1.E-14
      G(1)=((2.*R(3,13)-R(3,12)-R(3,14))/R(3,13)+(2*R(4,12)-R(4,11)-R(4,
     113))/R(4,12)+(2*R(5,11)-R(5,10)-R(5,12))/R(5,11))/6.0*100.0
      G(2)=((2*R(3,13)-R(2,13)-R(4,13))/R(3,13)+(2*R(4,12)-R(3,12)-R(5,1
     12))/R(4,12)+(2*R(5,11)-R(4,11)-R(6,11))/R(5,11))/6.0*100.0
      G(3)=((R(5,7)-R(5,6))/R(5,7)+(R(5,9)-R(5,10))/R(5,9))/2.0*100.0
      G(4)=((R(5,7)-R(4,7))/R(5,7)+(R(5,8)-R(4,8))/R(5,8)+(R(5,9)-R(4,9)
     1)/R(5,9))/3.0*100.0
      G(5)=((R(6,6)-R(6,5))/R(6,6)+(R(6,10)-R(6,11))/R(6,10)+(R(7,6)-R(7
     1,5))/R(7,6)+(R(7,10)-R(7,11))/R(7,10))/4.0*100.0
      G(6)=((R(6,6)-R(5,6))/R(6,6)+(R(6,10)-R(5,10))/R(6,10))/2.0*100.0
      G(7)=((R(8,4)-R(8,3))/R(8,4)+(R(8,13)-R(8,14))/R(8,13)+(R(9,4)-R(9
     1,3))/R(9,4)+(R(9,13)-R(9,14))/R(9,13)+(R(10,4)-R(10,3))/R(10,4)+(R
     2(10,13)-R(10,14))/R(10,13)+(R(11,4)-R(11,3))/R(11,4)+(R(11,13)-R(1
     31,14))/R(11,13))/8.0*100.0
      G(8)=((R(8,4)-R(7,4))/R(8,4)+(R(8,5)-R(7,5))/R(8,5)+(R(8,11)-R(7,1
     11))/R(8,11)+(R(8,12)-R(7,12))/R(8,12)+(R(8,13)-R(7,13))/R(8,13)+(R
     2(11,4)-R(12,4))/R(11,4)+(R(11,13)-R(12,13))/R(11,13))/7.0*100.0
      G(9)=((R(12,5)-R(12,4))/R(12,5)+(R(12,12)-R(12,13))/R(12,12))*50.0
      G(10)=((R(12,5)-R(13,5))/R(12,5)+(R(12,6)-R(13,6))/R(12,6)+(R(12,7
     1)-R(13,7))/R(12,7)+(R(12,8)-R(13,8))/R(12,8)+(R(12,9)-R(13,9))/R(1
     22,9)+(R(12,10)-R(13,10))/R(12,10)+(R(12,11)-R(13,11))/R(12,11)+(R(
     312,12)-R(13,12))/R(12,12))/8.0*100.0
      G(11)=(G(1)+G(3)+G(5)+G(7)+G(9))/5.
      G(12)=(G(2)+G(4)+G(6)+G(8)+G(10))/5.
      RETURN
      END
      SUBROUTINE HAAR(Q,E,NU)
      DIMENSION T(16),Q(30),E(5)
      N =2**NU
      NO =N
      ME = NU+2
      IP = (-2)*N
      IR = 0
```

```
 5 NO = NO/2
   IP = IP + 4*NO
   IR = IR + 2*NO
   ME = ME-1
   DO 6 J =1,NO
   L = 2*J
   K = L-1
   NOJ = NO+J
   T(NOJ) = E(ME)*(Q(IP+K)-Q(IP+L))
 6 Q(IR+J) =Q(IP+K)+Q(IP+L)
   IF (NO.GT.2) GO TO 5
   T(2) =E(2)*(Q(IR+1)-Q(IR+2))
   T(1) =E(1)*(Q(IR+1)+Q(IR+2))
   Q(1) =(T(1)+T(2))/FLOAT(N)
   Q(2) =(T(1)-T(2))/FLOAT(N)
   IP= -1
   IR = 0
 7 N2 = NO
   X = FLOAT(N2)/FLOAT(N)
   NO = NO*2
   IP = IP + NO/4
   IR = IR + NO/2
   DO 8 J =1,NO
   ND = (J+1)/2
   ID = N2 + ND
   Y = (-1.0)**(J+1)*X
 8 Q(IR+J) = Q(IP+ND)+Y*T(ID)
   IF (NO.LT.N) GO TO 7
   RETURN
   END
   SUBROUTINE HAR16H(XIN,HM)
   DIMENSION XIN(16,16),Q(30),HM(8),E(5)
   DO 1 I = 1,5
 1 E(I) = HM(I)
   NU = 4
   M = 14
   DO 3 I = 1,16
   DO 4 J = 1,16
 4 Q(J) = XIN(I,J)
   CALL HAAR(Q,E,NU)
   DO 5 K = 1,16
 5 XIN(I,K)=Q(M+K)
 3 CONTINUE
   RETURN
   END
```

APPENDIX C. PROGRAM STN

```
      PROGRAM STN(INPUT,OUTPUT,TAPE7)
      DIMENSION NV(9),IC(9),XIN(16,16),SIGI(7),XNI(6),SNI(10),XM(144,8,2
     &),HM(8),VM(8),SIGF(7,16,9),XNF(6,16,9),SNF(10,16,9),IWORK(2,10)
      DATA NV/5,5,5,3,4,8,8,8,8/
      DATA IC/"H","V","H","H","H","H","B","H","B"/
      DATA SIGI/9.,10.,11.,12.,14.,16.,18./
      DATA XNI/2.,3.,4.,5.,6.,10./
      DATA SNI/1.5,2.,2.5,3.,3.5,4.,4.5,5.,5.5,6./
      DATA NSIG,NNOI,NSTN/7,6,10/
      DATA IWORK/1,5,2,4,2,3,1,2,5,3,6,3,7,3,2,1,3,1,4,1/
      CALL SECOND(A)
      CALL RANSET(1)
      PRINT *,"ENTER MIN & MAX ALG. NO.S, NO. SETS, & NO. RUNS/SET. "
      READ *,IALG1,IALG2,MSET,NRUN
      DO 1 IALG = IALG1,IALG2
      CALL IPICK(IALG,IC,I1,I2)
      N = NV(IALG)
      DO 1 ISET = 1,MSET
      IUSE = MSET * (IALG-1) + ISET
      DO 1 IKIND = I1,I2
      PRINT *,"ENTER ",N," ALG ",IALG,IC(IALG)," M VALUES, SET ",ISET
      READ *,(XM(IUSE,J,IKIND),J=1,N)
    1 PRINT *,"VERIFICATION, ",(XM(IUSE,J,IKIND),J=1,N)," SET ",ISET
      DO 2 IALG = IALG1,IALG2
      CALL IPICK(IALG,IC,I1,I2)
      N = NV(IALG)
      DO 2 ISET = 1,MSET
      IUSE = MSET * (IALG-1) + ISET
      DO 3 IKIND = I1,I2
      DO 3 J = 1,N
      IF(IKIND.EQ.1) HM(J) = XM(IUSE,J,IKIND)
    3 IF(IKIND.EQ.2) VM(J) = XM(IUSE,J,IKIND)
      DO 4 ISIG = 1,NSIG
      SIG = SIGI(ISIG)
      DO 5 I = 1,16
      DO 5 J = 1,16
    5 XIN(I,J) = 0.0
      DO 6 I = 1,3
      XIN(2+I,14-I) = SIG
    6 XIN(5,6+I) = SIG
      DO 7 I = 6,7
      DO 7 J = 6,10
    7 XIN(I,J) = SIG
      DO 8 I = 8,11
      DO 8 J = 4,13
    8 XIN(I,J) = SIG
      DO 9 J = 5,12
    9 XIN(12,J) = SIG
```

```
      CALL SPICK(IALG,XIN,HM,VM)
      CALL APSIG(XIN,PSIG)
    4 SIGF(ISIG, ISET, IALG) = PSIG
      DO 10 INOI = 1,NNOI
      RMS = XNI(INOI)
      SQ = RMS*SQRT(33./96.)
      C = 2. * RMS**2 /SQ
      SUMNOI = 0.0
      DO 11 IRUN = 1,NRUN
      DO 12 I = 1,16
      DO 12 J = 1,16
   12 XIN(I,J) = C* (RANF(9) - 0.5)
      CALL SPICK(IALG,XIN,HM,VM)
      CALL PRMS(XIN,CRMS)
   11 SUMNOI = SUMNOI + CRMS
      SUMNOI = SUMNOI / FLOAT(NRUN) + 1.E-14
   10 XNF(INOI,ISET,IALG) = SUMNOI
      DO 2 ISTN = 1,NSTN
      N1 = IWORK(1,ISTN)
      N2 = IWORK(2,ISTN)
    2 SNF(ISTN,ISET,IALG) = SIGF(N1,ISET,IALG)/XNF(N2,ISET,IALG)
      DO 13 IALG = IALG1,IALG2
      CALL IPICK(IALG,IC,I1,I2)
      N = NV(IALG)
      WRITE (7,100) IALG,MSET,NRUN
  100 FORMAT(1H ,"ALG ",I2,",SETS ",I2,", RUNS ",I3)
      DO 14 ISET = 1,MSET
      IUSE = MSET * (IALG-1) + ISET
      DO 14 IKIND = I1,I2
   14 WRITE(7,200) ISET,(XM(IUSE,J,IKIND),J=1,N)
  200 FORMAT(1H ,I2,8F5.1)
      DO 15 I = 1,NSIG
   15 WRITE(7,300) SIGI(I),(SIGF(I,J,IALG),J=1,MSET)
      DO 16 I = 1,NNOI
   16 WRITE(7,300) XNI(I),(XNF(I,J,IALG),J=1,MSET)
      DO 13 I = 1,NSTN
   13 WRITE(7,300) SNI(I),(SNF(I,J,IALG),J=1,MSET)
  300 FORMAT(1H ,F4.1,4F10.5,3(/5X,4F10.5))
      END
      SUBROUTINE SPICK(IALG,XIN,HM,VM)
      DIMENSION XIN(16,16),HM(8),VM(8)
      IF(IALG .EQ. 1) CALL HAR16H(XIN,HM)
      IF(IALG .EQ. 2) CALL HAR16V(XIN,VM)
      IF(IALG .EQ. 3) CALL HAR16HV(XIN,HM,VM)
      IF(IALG .EQ. 4) CALL HAR4H(XIN,HM)
      IF(IALG .EQ. 5) CALL HAD4H(XIN,HM)
      IF(IALG .EQ. 6) CALL FT16H(XIN,HM)
      IF(IALG .EQ. 7) CALL FT16D2(XIN,HM,VM)
      IF(IALG .EQ. 8) CALL FFTH(XIN,HM)
      IF(IALG .EQ. 9) CALL FFTHV(XIN,HM,VM)
      RETURN
```

APPENDIX D. PROGRAM FRAME

```
      PROGRAM FRAME(TAPE1=/1360,TAPE2=/1360,OUTPUT)
      DIMENSION INBUF(335),IOUTBF(320),FRM(128,320),Q(510),
     &NPH(2),NPV(4),NBH(2),NBV(4),EHOR(9),EVERT(9),EH(9,2),EV(9,4)
      DATA NPH,NPV,NBH,NBV/256,64,3*128,32,6*1/
      DATA EH/5*.25,4*1.,3*.25,4*1.,2*0./
      DATA EV/4*.25,4*1.,0.,4*.25,4*1.,0.,4*.25,4*1.,0.,2*.25,4*1.,3*0./
      CALL SECOND(A1)
      CALL RANSET(A1)
      ISHIFT = 7
      LINDRP = 2
      RMS = 0.0
      SQ = RMS * SQRT((2.*320.+1.)/(6.*320.))
      IF(RMS .EQ. 0.) CN = 0.
      IF(RMS .NE. 0. ) CN = 2. * RMS**2 / SQ
C-FOR PROPER SCALING OF NOISE LEVEL.
C-READ EACH OF 43 420 LINE X 335 POINT/LINE IMAGES.
C-2 SECTIONS EACH OF 416 X 320 FRAME IN 3 HOR. BELTS AS FOLLOWS:
C-3 BELTS OF 128 LINES X 256 COL. & 128 LINES X  64 COL.
C-1 BELT  OF  32 LINES X 256 COL. &  32 LINES X  64 COL.
      DO 1 IMAGE = 1,43
      IGMIN = 100000000
      IGMAX = -1*IGMIN
      PRINT *,"READY TO BEGIN PROCESSING IMAGE ",IMAGE
      DO 2 LINSKP = 1,LINDRP
    2 READ (1,100) L,INBUF
  100 FORMAT (I5,335Z4)
      DO 3 IBELT = 1,4
      NROWB = NPV(IBELT)
      NBLOKV = NBV(IBELT)
      NPBV = NROWB/NBLOKV
      NUPBV = ALOG(FLOAT(NPBV)) / ALOG(2.) + .1
C-NO. & LOG TO BASE 2 POINTS IN VERTICAL BLOCK.
      NUPBV1 = NUPBV + 1
      MV = NPBV -2
      DO 4 LINE = 1,NROWB
      READ (1,100) L,INBUF
      DO 4 IPOINT = 1,320
    4 FRM(LINE,IPOINT) = INBUF(IPOINT+ISHIFT) + CN*(RANF(9)-.5)
      DO 5 I = 1,NUPBV1
    5 EVERT(I) = EV(I,IBELT)
C-VERTICAL ENHANCER ARRAYS.
      IHBASE = 0
      DO 6 ISEC = 1,2
      NCOLS = NPH(ISEC)
      NBLOKH = NBH(ISEC)
      NPBH = NCOLS / NBLOKH
      NUPBH = ALOG(FLOAT(NPBH))/ ALOG(2.) + .1
C-NO. AND LOG TO BASE 2 POINTS IN HORIZONTAL BLOCK.
```

```
      NUPBH1 = NUPBH + 1
      MH = NPBH -2
      IF(ISEC.GT.1) IHBASE=IHBASE+NPH(ISEC-1)
      DO 7 I = 1,NUPBH1
    7 EHOR(I) = EH(I,ISEC)
C-HORIZONTAL ENHANCER ARRAYS.
C-SCAN AND PROCESS HORIZONTALLY:
      DO 8 IROW = 1,NROWB
      DO 8 IBLOCK = 1,NBLOKH
      DO 9 IPOINT = 1,NPBH
      IHPOS = IHBASE + NPBH*(IBLOCK-1)+IPOINT
    9 Q(IPOINT) = FRM(IROW,IHPOS)
      CALL HAAR(Q,EHOR,NUPBH)
      DO 8 IPOINT = 1,NPBH
      IHPOS = IHBASE + NPBH*(IBLOCK-1) + IPOINT
    8 FRM(IROW,IHPOS) = Q(MH+IPOINT)
C-SCAN VERTICALLY AND PROCESS:
      DO 6 ICOL = 1,NCOLS
      DO 6 IBLOCK = 1,NBLOKV
      DO 10 IPOINT = 1,NPBV
      IVPOS = NPBV*(IBLOCK-1) + IPOINT
   10 Q(IPOINT) = FRM(IVPOS,ICOL)
      CALL HAAR(Q,EVERT,NUPBV)
      DO 6 IPOINT = 1,NPBV
      IVPOS = NPBV*(IBLOCK-1) + IPOINT
    6 FRM(IVPOS,ICOL) = Q(MV+IPOINT)
      DO 3 LINE = 1,NROWB
      DO 11 IPOINT = 1,320
      IOUTBF(IPOINT) = FRM(LINE,IPOINT) + .5
      IF (IOUTBF(IPOINT).GT. IGMAX) IGMAX = IOUTBF(IPOINT)
   11 IF(IOUTBF(IPOINT) .LT. IGMIN) IGMIN = IOUTBF(IPOINT)
C-READING THE LINES TO THE OUTPUT BUFFER FOR TAPING.
    3 WRITE (2,100) LINE,IOUTBF
   12 READ(1,100) L,INBUF
      IF(EOF(1) .EQ. 0.0) GO TO 12
C-STATEMENT LABELED 12 INSURES INPUT EOF REACHED.
      ENDFILE 2
      PRINT *,"FINISHED PROCESSING IMAGE ",IMAGE
    1 PRINT *,"IGMIN = ",IGMIN,", IGMAX = ",IGMAX
C-INSURE EOF(2) REACHED.
      PRINT *,"HAVE REACHED THE END OF THE PROGRAM."
      END
)E
```

DISTRIBUTION LIST

Director
U.S. Army Research Office
ATTN: Library
P. O. Box 12211
Research Triangle Park, NC 27709

Director
Department of Defense
Advanced Research and Projects Agency
Washington, DC 20301

Commander
U.S. Army Materiel Development and
Readiness Command
ATTN: DRCSA-R
DRCSA-PP
Washington, DC 20315

Commander
U.S. Army Aviation Systems Command
12th & Spruce Streets
St. Louis, MO 63166

Commander
U.S. Army Tank-Automotive Research and
Development Command
ATTN: DRDTA, Technical Library
Warren, MI 48090

Commander
U.S. Army Electronics Command and
Devices Laboratory
Ft. Monmouth, NJ 07703

Commander
U.S. Army Electronics Command
Atmospheric Science Laboratory
White Sands Missile Range
ATTN: DRSEL-BL-MS
Mr. Richard Gomez
White Sands, NM 88002

Commander
U.S. Army Test and Evaluation Command
ATTN: AMSTE-TA-A (2)
STEAP-TL
STEAP-DS-LP
Aberdeen Proving Ground, MD 21005

Commander
U.S. Army Harry Diamond Laboratories
ATTN: AMXDO-TD/002
AMXDO-TIB
2800 Powder Mill Road
Adelphi, MD 20783

Commander
U.S. Army Armament Research and Development Command
ATTN: DRDAR-SC, COL A. Larkins
DRDAR-SC, Dr. D. A. Gyorog
DRDAR-SCF, L. Berman
DRDAR-SCF-I, J. Lehman
DRDAR-SCF-IO, L. Lester
DRDAR-SCF-IO, G. Sivak (12)
DRDAR-SCP, J. Glennon
DRDAR-MSA, R. Isakower
DRDAR-TSS (5)
Dover, NJ 07801

Defense Technical Information Center (12)
Cameron Station
Alexandria, VA 22314

Commander
U.S. Army Armament Materiel and
Readiness Command
ATTN: DRDAR-LEP-L
Rock Island, IL 61299

Director
U.S. Army TRADOC Systems
Analysis Activity
ATTN: ATAA-SL (Tech Lib)
White Sands Missile Range, NM 88002

U.S. Army Materiel Systems
Analysis Activity
ATTN: DRXSY-MP
Aberdeen Proving Ground, MD 21005

Weapon System Comcept Team/CSL
ATTN: DRDAR-ACW
Aberdeen Proving Ground, MD 21010

Technical Library
ATTN: DRDAR-CLJ-L
Aberdeen Proving Ground, MD 21010

Director
U.S.Army Ballistic Research Laboratory
ARRADCOM
ATTN: DRDAR-TSB-S
Aberdeen Proving Ground, MD 21005

Technical Library
ATTN: DRDAR-LCB-TL
Benet Weapons Laboratory
Watervliet, NY 12189

Dr. Thomas S. Huang
Institute for Image Technology
1000 North Western Ave.
West Lafayette, Indiana 47906